

Magnet Project API Documentation

Route Authentication & Role Protection Summary

Module	Route/Action	Public	Auth Required	Role(s) Required
Products	GET /api/products	No	Yes	Any authenticated user
	GET /api/products/:id	No	Yes	Any authenticated user
	POST /api/products/addProductsByBusiness	No	Yes	business
	POST /api/products/addProductsByMagnet_employee	No	Yes	magnet_employee, admin
	PUT /api/products/:id	No	Yes	business (own), admin, magnet_employee
	DELETE /api/products/:id	No	Yes	business (own), admin, magnet_employee
	PUT /api/products/:id/approve	No	Yes	admin, magnet_employee
	PUT /api/products/:id/decline	No	Yes	admin, magnet_employee
Categories	GET /api/categories	Yes	No	None
	POST /api/categories	No	Yes	business, admin, magnet_employee
	PUT /api/categories/:id	No	Yes	business (own), admin, magnet_employee
	DELETE /api/categories/:id	No	Yes	business (own), admin, magnet_employee
Orders	GET /api/orders/status-options	Yes	No	None
	POST /api/orders	No	Yes	customer
	GET /api/orders/my	No	Yes	customer
	GET /api/orders/business-products	No	Yes	business

Module	Route/Action	Public	Auth Required	Role(s) Required
	GET /api/orders/:id	No	Yes	admin, magnet_employee, customer (own)
	GET /api/orders	No	Yes	admin, magnet_employee
	PUT /api/orders/:id/status	No	Yes	admin, magnet_employee
Reviews	POST /api/reviews/products/:id/reviews	No	Yes	customer
	GET /api/reviews/products/:id/reviews	Yes	No	None
	GET /api/reviews/business-products-reviews	No	Yes	business
	DELETE /api/reviews/:id	No	Yes	admin, magnet_employee
Wishlist	GET /api/wishlist	No	Yes	customer
	POST /api/wishlist	No	Yes	customer
	DELETE /api/wishlist/:productId	No	Yes	customer
Address	GET /api/addresses	No	Yes	customer
	POST /api/addresses	No	Yes	customer
	PUT /api/addresses/:id	No	Yes	customer
	DELETE /api/addresses/:id	No	Yes	customer
User	GET /api/user/profile	No	Yes	customer
	PUT /api/user/profile	No	Yes	customer
	GET /api/user/business-requests	No	Yes	admin, magnet_employee
	POST /api/user/business-approval	No	Yes	admin, magnet_employee
	GET /api/user/business/:businessId	No	Yes	admin, magnet_employee
	GET /api/user/business-profile	No	Yes	business, admin, magnet_employee
Admin	POST /api/admin/users	No	Yes	admin
	GET /api/admin/users	No	Yes	admin
	GET /api/admin/users/stats	No	Yes	admin

Module	Route/Action	Public	Auth Required	Role(s) Required
	GET /api/admin/users/:id	No	Yes	admin
	PUT /api/admin/users/:id	No	Yes	admin
	DELETE /api/admin/users/:id	No	Yes	admin
	PUT /api/admin/users/:id/disallow	No	Yes	admin
	PUT /api/admin/users/:id/allow	No	Yes	admin
	PUT /api/admin/users/:id/verify-email	No	Yes	admin, magnet_employee
	PUT /api/admin/users/:id/unverify-email	No	Yes	admin, magnet_employee
	PUT /api/admin/users/:id/verify-phone	No	Yes	admin, magnet_employee
	PUT /api/admin/users/:id/unverify-phone	No	Yes	admin, magnet_employee
	GET /api/admin/wishlists	No	Yes	admin
	GET /api/admin/wishlists/:id	No	Yes	admin
	POST /api/admin/wishlists	No	Yes	admin
	PUT /api/admin/wishlists/:id	No	Yes	admin
	DELETE /api/admin/wishlists/:id	No	Yes	admin
	GET /api/admin/reviews	No	Yes	admin
	GET /api/admin/reviews/:id	No	Yes	admin
	POST /api/admin/reviews	No	Yes	admin
	PUT /api/admin/reviews/:id	No	Yes	admin
	PUT /api/admin/reviews/:id/reject	No	Yes	admin

|| DELETE /api/admin/reviews/:id | No | Yes | admin || | GET /api/admin/addresses | No | Yes | admin || | GET /api/admin/addresses/:id | No | Yes | admin || | POST /api/admin/addresses | No | Yes | admin || | PUT /api/admin/addresses/:id | No | Yes | admin || | DELETE /api/admin/addresses/:id | No | Yes | admin || | GET /api/admin/orders | No | Yes | admin || | GET /api/admin/orders/:id | No | Yes | admin || | POST /api/admin/orders | No | Yes | admin || | PUT /api/admin/orders/:id | No | Yes | admin || | DELETE /api/admin/orders/:id | No | Yes | admin || Auth | Registration/Login/OTP | Yes | No | None || | Forgot Password | No | Yes | Any authenticated user |

Notes

- **Auth Required:** Requires a valid JWT token in the `Authorization` header.
- **Role(s) Required:** Requires the user to have one of the listed roles. If "(own)" is specified, the user must own the resource (e.g., their own product or order).
- **Public:** Anyone can access, no authentication required.
- **All:** All HTTP methods (GET, POST, PUT, DELETE) for that endpoint.
- **Business (if owns product in order):** Business user can access if any product in the order is owned by them (see order tracking section).
- **Enhanced Data Population:** All APIs now return complete user information instead of just IDs for related fields (owner, approvedBy, createdBy, customer, user, etc.).

NEW Order Total Calculation Feature

All order-related endpoints now include automatic total calculation:

New Fields in Order Responses

- `total` (number): Overall order total (sum of all item totals)
- `itemTotal` (number): Individual item total for each order item (quantity × pricePerUnit)

Affected Endpoints

All order routes in both customer and admin APIs now include these calculated fields:

- **Customer Order Routes:** `/api/orders/*`
- **Admin Order Routes:** `/api/admin/orders/*`
- **Business Order Routes:** `/api/orders/business-products`

Calculation Details

- Server-side calculation ensures consistency
- Prices parsed from string format to numbers
- Invalid/missing prices default to 0
- Works with all language preferences (`lang=en`, `lang=ar`, `lang=both`)

Backward Compatibility

- Existing API structure unchanged
- New fields added without breaking changes
- All existing functionality preserved

Message Format

All API responses return bilingual messages in both English and Arabic for every endpoint, including all errors and successes. The system uses a centralized `messages.js` file to ensure consistency across all modules (auth, user, product, order, address, category, review, wishlist, etc.).

Example:

```
{  
  "status": "success",  
  "message": {  
    "en": "User registered successfully",  
    "ar": "تم تسجيل المستخدم بنجاح"  
  },  
  "data": { ... }  
}
```

Enhanced User Data Population

All APIs now return complete user information instead of just IDs for related fields. This includes:

User Object Format

When user data is populated, it returns:

```
{  
  "id": "user_id",  
  "firstname": "John",  
  "lastname": "Doe",  
  "email": "john@example.com",  
  "role": "customer"  
}
```

Business User Object Format

For business users, additional company information is included:

```
{  
  "id": "user_id",  
  "firstname": "Business",  
  "lastname": "Owner",  
  "email": "business@example.com",  
  "role": "business",  
  "companyName": "ABC Company"  
}
```

Admin User Object Format

For admin users who approved content:

```
{  
  "id": "admin_id",  
  "firstname": "Admin",  
  "lastname": "Approved",  
  "email": "admin@example.com",  
  "role": "admin",  
  "approvedContent": true  
}
```

```
"lastname": "User",
"email": "admin@example.com",
"role": "admin"
}
```

Populated Fields

The following fields are now populated with complete user objects instead of IDs:

- **Product Owner:** `product.owner` - Returns business user details
- **Product Approved By:** `product.approvedBy` - Returns admin/employee details
- **Category Created By:** `category.createdBy` - Returns creator details
- **Review User:** `review.user` - Returns reviewer details
- **Order Customer:** `order.customer` - Returns customer details
- **Address User:** `address.user` - Returns address owner details
- **Wishlist User:** `wishlist.user` - Returns wishlist owner details
- **Business Approved By:** `business.businessInfo.approvedBy` - Returns approver details

Example Enhanced Responses

Product Response

```
{
  "status": "success",
  "data": {
    "product": {
      "id": "product_id",
      "name": "Product Name",
      "code": "PROD001",
      "owner": {
        "id": "owner_id",
        "firstname": "Business",
        "lastname": "Owner",
        "email": "business@example.com",
        "companyName": "ABC Company"
      },
      "approvedBy": {
        "id": "admin_id",
        "firstname": "Admin",
        "lastname": "User",
        "email": "admin@example.com",
        "role": "admin"
      }
    }
  }
}
```

Review Response

```
{  
  "status": "success",  
  "data": {  
    "review": {  
      "id": "review_id",  
      "rating": 5,  
      "comment": "Great product!",  
      "user": {  
        "id": "user_id",  
        "firstname": "John",  
        "lastname": "Doe",  
        "email": "john@example.com",  
        "role": "customer"  
      },  
      "product": {  
        "id": "product_id",  
        "name": "Product Name",  
        "owner": {  
          "id": "owner_id",  
          "firstname": "Business",  
          "lastname": "Owner",  
          "email": "business@example.com",  
          "companyName": "ABC Company"  
        }  
      }  
    }  
  }  
}
```

Order Response

```
{  
  "status": "success",  
  "data": {  
    "order": {  
      "id": "order_id",  
      "status": "pending",  
      "customer": {  
        "id": "customer_id",  
        "firstname": "John",  
        "lastname": "Doe",  
        "email": "john@example.com",  
        "role": "customer"  
      },  
      "items": [  
        {  
          "product": {  
            "id": "product_id",  
            "name": "Product Name",  
            "pricePerUnit": "25.50",  
            "quantity": 2  
          }  
        }  
      ]  
    }  
  }  
}
```

```

    "owner": {
        "id": "owner_id",
        "firstname": "Business",
        "lastname": "Owner",
        "email": "business@example.com",
        "companyName": "ABC Company"
    },
    "quantity": 2,
    "itemTotal": 51.00
},
],
"total": 51.00,
"createdAt": "2024-01-01T00:00:00.000Z",
"updatedAt": "2024-01-01T00:00:00.000Z"
}
}
}
}

```

Order Total Calculation

All order responses now include automatic total calculation:

Total Fields

- **total** (number): Overall order total (sum of all item totals)
- **itemTotal** (number): Individual item total (quantity × pricePerUnit) for each item
- **pricePerUnit** (string): Product price per unit shown in product object within each order item

Calculation Logic

- Each item's **itemTotal** = **quantity** × **pricePerUnit**
- Order **total** = sum of all item totals
- Prices are parsed from string format to numbers for calculation
- Invalid or missing prices default to 0
- All calculations are performed server-side for consistency

Example Calculation

```
{
  "items": [
    {
      "product": { "pricePerUnit": "25.50" },
      "quantity": 2,
      "itemTotal": 51.00 // 25.50 × 2
    },
    {
      "product": { "pricePerUnit": "15.00" },
      "quantity": 1,
      "itemTotal": 15.00 // 15.00 × 1
    }
  ],
  "total": 66.00
}
```

```
        "itemTotal": 15.00 // 15.00 × 1
    }
],
"total": 66.00 // 51.00 + 15.00
}
```

Language Support

- Total calculations work with all language preferences (`lang=en`, `lang=ar`, `lang=both`)
 - Numeric values remain consistent across languages
 - Currency formatting is handled client-side based on locale
-

Verification Rules

Phone Number Verification

- **If phone number is provided:** Only the phone is verified (`isPhoneVerified = true`, `isEmailVerified = false`)
- **If no phone number is provided:** Only the email is verified (`isEmailVerified = true`, `isPhoneVerified = false`)

This applies to both customer and business registration.

Validation Rules

Email Validation

- **Format:** Must match email format: `^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$`
- **Required:** Email is required for all user registrations
- **Unique:** Email must be unique across all users

Phone Validation

- **Format:** Must match phone format: `^\+?[1-9]\d{1,14}$`
- **Optional:** Phone number is optional for customer registration
- **Unique:** If provided, phone must be unique across all users
- **Saudi Numbers:** Saudi phone numbers are automatically verified

Password Requirements

- **Minimum Length:** 8 characters
- **Complexity:** Must contain at least:
 - One uppercase letter (A-Z)
 - One lowercase letter (a-z)
 - One number (0-9)
- **Pattern:** `^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)`

Name Validation

- **First Name:** 2-50 characters, required
- **Last Name:** 2-50 characters, required
- **Trim:** Leading and trailing whitespace is automatically removed

Business Registration Validation

- **CR Number:** 1-50 characters, required
- **VAT Number:** 1-50 characters, required
- **Company Name:** 2-100 characters, required
- **Company Type:** 2-100 characters, required
- **Country:** 2-50 characters, required
- **City:** 2-50 characters, required
- **District:** 2-50 characters, required
- **Street Name:** 2-100 characters, required

Product Validation

- **Category:** Required bilingual object { en: "English Category", ar: "Arabic Category" }
- **Name:** Required bilingual object { en: "English Name", ar: "Arabic Name" }
- **Description:** Optional bilingual object { en: "English Description", ar: "Arabic Description" }
- **Unit:** Optional bilingual object { en: "English Unit", ar: "Arabic Unit" }
- **Custom Fields:** Array of objects, minimum 3, maximum 10
 - Each object must have **key** and **value** properties
 - Both key and value must be bilingual objects: { en: "English", ar: "Arabic" }
- **Code:** Optional, auto-generated if not provided
- **Images:** Optional array of strings (URLs)
- **Attachments:** Optional array of product IDs
- **Min Order:** Optional number
- **Price Per Unit:** Optional string
- **Stock:** Optional number

Address Validation

- **Address Line 1:** Required string
- **Address Line 2:** Optional string
- **City:** Required string
- **State:** Required string
- **Postal Code:** Required string
- **Country:** Required string

Category Validation

- **Name:** Required bilingual object { en: "English", ar: "Arabic" }
- **Description:** Optional bilingual object { en: "English", ar: "Arabic" }

Review Validation

- **Rating:** Required number, min: 1, max: 5

- **Comment:** Optional string

Admin User Management Validation

- **Role:** Must be one of: 'admin', 'magnet_employee', 'business', 'customer'
- **Business Info:** Required for business users
 - **CR Number:** 1-50 characters, required for business users
 - **VAT Number:** 1-50 characters, required for business users
 - **Company Name:** 2-100 characters, required for business users
 - **Company Type:** 2-100 characters, required for business users
 - **City:** 2-50 characters, required for business users
 - **District:** 2-50 characters, required for business users
 - **Street Name:** 2-100 characters, required for business users
- **Disallow Reason:** Required when disallowing a user
- **Self-Protection:** Admins cannot disallow or delete their own accounts
- **Admin Protection:** Admin users cannot be disallowed by other admins

Order Validation

- **Customer:** Required user ID
- **Items:** Required array of order items
 - Each item must have **product** (product ID) and **quantity** (number)
- **Shipping Address:** Required address ID
- **Status:** Optional, defaults to 'pending', enum: 'pending', 'confirmed', 'shipped', 'delivered', 'cancelled' (English values only, backend converts to bilingual)
 - **Valid English Statuses:** ['pending', 'confirmed', 'shipped', 'delivered', 'cancelled']
 - **Valid Arabic Statuses:** ['قيد الانتظار', 'مؤكد', 'تم الشحن', 'تم التوصيل', ['ملغي']]
 - **Validation:** Backend validates status using `Order.isValidStatus()` method
 - **Conversion:** Backend converts English status to bilingual using `Order.convertStatusToBilingual()` method

Auth Routes

1. Register User

- **POST /api/auth/register**
- **Description:** Register a new customer user with automatic verification based on phone number presence.
- **Body:**
 - **firstname** (string, required)
 - **lastname** (string, required)
 - **email** (string, required)
 - **phone** (string, optional)
 - **password** (string, required)
 - **country** (string, required)

- `language` (string, optional, default: 'en')
 - **Verification Logic:**
 - If `phone` is provided: `isPhoneVerified = true, isEmailVerified = false`
 - If `phone` is not provided: `isEmailVerified = true, isPhoneVerified = false`
 - **Response:**
 - `201 Created`: User info + JWT token with verification status
-

2. Register Business

- **POST** `/api/auth/business-register`
 - **Description:** Register a new business user (requires approval) with automatic verification based on phone number presence.
 - **Body:**
 - `firstname, lastname, email, password, crNumber, vatNumber, companyName, companyType, country, city, district, streetName, phone` (all required)
 - **Verification Logic:**
 - If `phone` is provided: `isPhoneVerified = true, isEmailVerified = false`
 - If `phone` is not provided: `isEmailVerified = true, isPhoneVerified = false`
 - **Response:**
 - `201 Created`: Business info (under review) with verification status
-

3. Send Email OTP

- **POST** `/api/auth/send-email-otp`
 - **Description:** Send an OTP to a new email (fails if email already exists).
 - **Body:**
 - `email` (string, required)
 - **Response:**
 - `200 OK`: Success message (bilingual)
-

4. Send Phone OTP

- **POST** `/api/auth/send-phone-otp`
 - **Description:** Send an OTP to a new phone (fails if phone already exists).
 - **Body:**
 - `phone` (string, required)
 - **Response:**
 - `200 OK`: Success message (bilingual)
-

5. Confirm OTP

- **POST** `/api/auth/confirm-otp`
- **Description:** Confirm an OTP for any email or phone identifier (works for both registered and non-registered users).
- **Body:**

- `identifier` (string, required)
 - `otp` (string, required)
 - **Response:**
 - `200 OK`: OTP verified successfully (bilingual message)
-

6. Confirm Login OTP

- **POST** `/api/auth/confirm-login-otp`
 - **Description:** Confirm an OTP for registered users and return user data with JWT token (like login API).
 - **Body:**
 - `identifier` (string, required)
 - `otp` (string, required)
 - **Response:**
 - `200 OK`: User info + JWT token (bilingual message)
 - `404 Not Found`: User not found (bilingual message)
-

7. Login

- **POST** `/api/auth/login`
 - **Description:** Login with email/phone and password.
 - **Body:**
 - `identifier` (email or phone, required)
 - `password` (string, required)
 - **Response:**
 - `200 OK`: User info + JWT token (bilingual message)
-

8. Login with OTP

- **POST** `/api/auth/login-with-otp`
 - **Description:** Request an OTP for login (email or phone).
 - **Body:**
 - `identifier` (email or phone, required)
 - **Response:**
 - `200 OK`: OTP sent (bilingual message)
-

9. Forgot Password

- **POST** `/api/auth/forgot-password`
 - **Description:** Change password for authenticated user.
 - **Headers:** `Authorization: Bearer <token>`
 - **Body:**
 - `newPassword` (string, required)
 - **Response:**
 - `200 OK`: Password updated (bilingual message)
-

10. Create Admin User

- **POST** `/api/auth/create-admin`
 - **Description:** Create a new admin user. Only accessible by authenticated admin users.
 - **Headers:** `Authorization: Bearer <token>`
 - **Body:**
 - `firstname` (string, required)
 - `lastname` (string, required)
 - `email` (string, required)
 - `phone` (string, optional)
 - `password` (string, required)
 - `country` (string, required)
 - `language` (string, optional, default: 'en')
 - **Response:**
 - `201 Created`: Admin user info (bilingual message)
 - `403 Forbidden`: Insufficient permissions (bilingual message)
 - `400 Bad Request`: Email/phone already registered (bilingual message)
-

11. Create Magnet Employee User

- **POST** `/api/auth/create-magnet-employee`
 - **Description:** Create a new magnet employee user. Only accessible by authenticated admin users.
 - **Headers:** `Authorization: Bearer <token>`
 - **Body:**
 - `firstname` (string, required)
 - `lastname` (string, required)
 - `email` (string, required)
 - `phone` (string, optional)
 - `password` (string, required)
 - `country` (string, required)
 - `language` (string, optional, default: 'en')
 - **Response:**
 - `201 Created`: Magnet employee user info (bilingual message)
 - `403 Forbidden`: Insufficient permissions (bilingual message)
 - `400 Bad Request`: Email/phone already registered (bilingual message)
-

Product Routes

Language Support

Products support bilingual content (Arabic and English). You can:

- **Get products in specific language:** Add `?lang=en` or `?lang=ar` to the URL
- **Get products in user's preferred language:** The system uses the user's language preference
- **Get full bilingual data:** Use `?lang=both` to get both languages

1. Get Products

- **GET /api/products**
- **GET /api/products?lang=en** (English only)
- **GET /api/products?lang=ar** (Arabic only)
- **GET /api/products?lang=both** (Both languages)
- **Description:** Get all approved products with pagination and filtering (requires authentication). Admin/magnet_employee can see all products.
- **Headers:** Authorization: Bearer <token>
- **Query Parameters:**
 - **page** (number, optional, default: 1) - Page number for pagination
 - **limit** (number, optional, default: 10) - Number of items per page
 - **category** (string, optional) - Filter by category (searches in English category name)
 - **search** (string, optional) - Search by product name (English/Arabic), code, or owner company name
 - **status** (string, optional) - Filter by status: 'pending', 'approved', 'declined' (admin/business/magnet_employee only)
- **Response:**
 - **200 OK**: List of products with pagination info
- **Product Object:**
 - **code** (string, auto-generated if not provided, e.g. "A001")
 - **category** (object, required, bilingual: { en: "English Category", ar: "Arabic Category" })
 - **name** (object, required, bilingual: { en: "English Name", ar: "Arabic Name" })
 - **images** (array of strings)
 - **description** (object, optional, bilingual: { en: "English Description", ar: "Arabic Description" })
 - **attachments** (array of product IDs, references to other products)
 - **unit** (object, optional, bilingual: { en: "English Unit", ar: "Arabic Unit" })
 - **minOrder** (number)
 - **pricePerUnit** (string)
 - **stock** (number)
 - **customFields** (array of objects, min 3, max 10, each: { key: { en: "English Key", ar: "Arabic Key" }, value: { en: "English Value", ar: "Arabic Value" } })
 - **status** (string: 'pending', 'approved', 'declined')
 - **owner** (object, populated with business user details)
 - **approvedBy** (object, populated with admin/employee details who approved)
 - **createdAt** (date)
- **Example Response:**

```
{
  "status": "success",
  "data": {
    "products": [
      {
        "id": "product_id",
        "code": "A001",
        "name": { "en": "Product Name", "ar": "اسم المنتج" },
        "category": { "en": "Electronics", "ar": "الكترونيات" },
        ...
      }
    ]
  }
}
```

```

        "status": "approved",
        "owner": {
            "id": "owner_id",
            "email": "business@example.com",
            "businessInfo": { "companyName": "ABC Company" }
        }
    }
],
{
    "pagination": {
        "currentPage": 1,
        "totalPages": 5,
        "totalItems": 50,
        "itemsPerPage": 10
    }
}
}

```

1a. Get Product by ID

- **GET** /api/products/:id
- **GET** /api/products/:id?lang=en (English only)
- **GET** /api/products/:id?lang=ar (Arabic only)
- **GET** /api/products/:id?lang=both (Both languages)
- **Description:** Get a single product by its ID. Requires authentication. Admin, business, and magnet_employee can access any product (including pending/declined).
- **Headers:** Authorization: Bearer <token>
- **Response:**
 - 200 OK: Product object (see above)
 - 404 Not Found: If product does not exist
 - 403 Forbidden: If trying to access a non-approved product without proper role

2. Add Product (Business)

- **POST** /api/products/addProductsByBusiness
- **Description:** Business user adds a new product (pending approval). Product code is auto-generated if not provided.
- **Headers:** Authorization: Bearer <token>
- **Body:**
 - category (object, required, bilingual: { en: "English Category", ar: "Arabic Category" })
 - name (object, required, bilingual: { en: "English Name", ar: "Arabic Name" })
 - images (array of strings, optional)
 - attachments (array of product IDs, references to other products, optional)
 - description (object, optional, bilingual: { en: "English Description", ar: "Arabic Description" })
 - unit (object, optional, bilingual: { en: "English Unit", ar: "Arabic Unit" })
 - minOrder, pricePerUnit, stock (all optional)

- `customFields` (array of objects, min 3, max 10, each: { `key`: { `en`: "English Key", `ar`: "Arabic Key" }, `value`: { `en`: "English Value", `ar`: "Arabic Value" } }, required)

- **Response:**

- `201 Created`: Product info (pending approval) with bilingual message
-

3. Add Product (Admin/Magnet Employee)

- **POST** `/api/products/addProductsByMagnet_employee`
 - **Description:** Admin or magnet employee adds a new product (approved immediately). Product code is auto-generated if not provided.
 - **Headers:** `Authorization: Bearer <token>`
 - **Body:**
 - `category` (object, required, bilingual: { `en`: "English Category", `ar`: "Arabic Category" })
 - `name` (object, required, bilingual: { `en`: "English Name", `ar`: "Arabic Name" })
 - `images` (array of strings, optional)
 - `attachments` (array of product IDs, references to other products, optional)
 - `description` (object, optional, bilingual: { `en`: "English Description", `ar`: "Arabic Description" })
 - `unit` (object, optional, bilingual: { `en`: "English Unit", `ar`: "Arabic Unit" })
 - `minOrder`, `pricePerUnit`, `stock` (all optional)
 - `customFields` (array of objects, min 3, max 10, each: { `key`: { `en`: "English Key", `ar`: "Arabic Key" }, `value`: { `en`: "English Value", `ar`: "Arabic Value" } }, required)
 - `owner` (user id, required)
 - **Response:**
 - `201 Created`: Product info (approved) with bilingual message
-

4. Update Product

- **PUT** `/api/products/:id`
 - **Description:** Business user updates their own product (pending approval). Admin/magnet_employee can update and approve/decline.
 - **Headers:** `Authorization: Bearer <token>`
 - **Body:**
 - Any product field (see above)
 - `status` (optional, only admin/magnet_employee can set to 'approved' or 'declined')
 - **Response:**
 - `200 OK`: Updated product info with bilingual message
-

5. Delete Product

- **DELETE** `/api/products/:id`
- **Description:** Business user deletes their own product, or admin/employee deletes any product.
- **Headers:** `Authorization: Bearer <token>`
- **Response:**

- 200 OK: Product deleted with bilingual message
-

6. Approve Product

- **PUT** /api/products/:id/approve
 - **Description:** Admin/magnet_employee approves a product.
 - **Headers:** Authorization: Bearer <token>
 - **Response:**
 - 200 OK: Product approved with bilingual message
-

7. Decline Product

- **PUT** /api/products/:id/decline
 - **Description:** Admin/magnet_employee declines a product.
 - **Headers:** Authorization: Bearer <token>
 - **Response:**
 - 200 OK: Product declined with bilingual message
-

Real-Time Order Tracking (WebSocket)

Overview

- The API supports real-time order status updates using WebSocket (Socket.IO).
- Clients can subscribe to updates for specific orders and receive instant notifications when the order status changes (e.g., confirmed, shipped, delivered).

How It Works

1. Client authenticates with a JWT token when connecting to the WebSocket.
2. Client joins a room for a specific order using the order ID.
3. When the order status changes, the server emits an orderStatusUpdate event to all clients in that room.

WebSocket Connection (Socket.IO)

- **URL:** ws://<your-server>:5000 (or wss:// for HTTPS)
- **Library:** [socket.io-client](#)

Client Example (JavaScript/React):

```
import { io } from 'socket.io-client';
const socket = io('http://localhost:5000', {
  auth: { token: '<JWT_TOKEN>' }
});

// Join the order room
socket.emit('joinOrderRoom', '<ORDER_ID>');
```

```
// Listen for updates
socket.on('orderStatusUpdate', (data) => {
  // data: { orderId, status, statusLog, updatedAt }
  console.log('Order update:', data);
});
```

Order Room Access Rules

- **Admin & magnet_employee:** Can track any order.
- **Business:** Can track orders that include products they own.
- **Customer:** Can track only their own orders.
- Unauthorized attempts to join order rooms will be ignored.

Server-Side Security

- The server authenticates each socket connection using the JWT token.
- The server enforces the above access rules for joining order rooms.

Order Status Update Event

- **Event:** orderStatusUpdate
- **Payload:**

```
{
  "orderId": "...",
  "status": "shipped",
  "statusLog": [
    { "status": "pending", "timestamp": "..." },
    { "status": "confirmed", "timestamp": "..." },
    { "status": "shipped", "timestamp": "..." }
  ],
  "updatedAt": "2024-07-10T12:00:00.000Z"
}
```

Security Notes

- Clients must provide a valid JWT token when connecting.
- The server will only allow joining order rooms for orders the user is authorized to track (see access rules above).
- Unauthorized attempts to join rooms will be ignored.

Additional Utility Features

Email and SMS Utilities

The API includes comprehensive email and SMS functionality for user verification and notifications:

Email Features

- **Email OTP Generation:** Automatic generation of 6-digit verification codes
- **Email Templates:** Pre-formatted HTML email templates for various notifications
- **Business Notifications:** Automatic email notifications for business registration status changes
- **Verification Emails:** Email verification for user registration

SMS Features

- **SMS OTP Generation:** Automatic generation of 6-digit SMS verification codes
- **Saudi Number Support:** Special handling for Saudi phone numbers (+966)
- **SMS Templates:** Pre-formatted SMS messages for various notifications
- **Phone Verification:** SMS verification for user registration

Product Code Generation

The API includes an automatic product code generation system:

Features

- **Auto-Generation:** If no product code is provided, the system automatically generates a unique code
- **Format:** Codes follow the pattern "A001", "A002", etc.
- **Uniqueness:** Each generated code is guaranteed to be unique
- **Custom Codes:** Users can provide their own custom codes if desired

User Model Features

The user model includes several advanced features:

Favorites System

- **Add to Favorites:** Users can add products to their favorites list
- **Remove from Favorites:** Users can remove products from their favorites
- **View Favorites:** Users can view their complete favorites list

Cart System

- **Add to Cart:** Users can add products to their shopping cart
- **Cart Management:** Users can update quantities and remove items
- **Cart Persistence:** Cart data is stored with the user account

Business Approval Workflow

- **Registration Review:** Business registrations require admin/employee approval
- **Status Tracking:** Business approval status is tracked (pending, approved, rejected)
- **Notification System:** Automatic notifications for approval status changes
- **Rejection Reasons:** Admins can provide reasons for business registration rejections

Verification Status Tracking

- **Email Verification:** Track whether user's email is verified
- **Phone Verification:** Track whether user's phone is verified
- **Automatic Verification:** Saudi phone numbers are automatically verified
- **Manual Verification:** Other verification methods available

User Disallowance System

- **Account Disallowance:** Admins can disallow users from logging in and performing actions
- **Disallowance Tracking:** Track who disallowed a user, when, and the reason
- **Allowance Tracking:** Track who allowed a user and when
- **Login Prevention:** Disallowed users cannot login or perform any actions
- **Business Action Prevention:** Disallowed business users cannot add products
- **Self-Protection:** Admins cannot disallow their own accounts
- **Admin Protection:** Admin users cannot be disallowed by other admins

Bilingual Message System

The API includes a comprehensive bilingual message system:

Features

- **Centralized Messages:** All messages are stored in a centralized `messages.js` file
- **Bilingual Support:** All messages are available in both English and Arabic
- **Consistent Format:** All API responses follow the same bilingual format
- **Error Messages:** Comprehensive error messages in both languages
- **Success Messages:** Success confirmations in both languages

Message Format

```
{  
  "status": "success",  
  "message": {  
    "en": "Operation completed successfully",  
    "ar": "تم إكمال العملية بنجاح"  
  },  
  "data": { ... }  
}
```

Authentication and Authorization

The API includes a robust authentication and authorization system:

JWT Authentication

- **Token-Based:** Uses JWT tokens for authentication
- **Expiration:** Tokens expire after 7 days
- **Refresh:** Tokens can be refreshed

- **Secure:** Uses environment variables for secret keys

Role-Based Access Control

- **Admin:** Full system access including user management
- **Magnet Employee:** Limited admin access
- **Business:** Business-specific features
- **Customer:** Customer-specific features

Admin User Management

- **User Creation:** Admins can create any type of user (customer, business, magnet_employee, admin)
- **User Management:** Complete CRUD operations on all users
- **User Statistics:** Comprehensive user statistics and analytics
- **User Disallowance:** Ability to disallow users from accessing the system
- **User Allowance:** Ability to re-allow previously disallowed users
- **Audit Trail:** Complete tracking of disallowance/allowance actions
- **Security Protections:** Self-protection and admin protection mechanisms

Middleware System

- **Auth Middleware:** Verifies JWT tokens
- **Role Middleware:** Enforces role-based access
- **Validation Middleware:** Validates request data
- **Error Handling:** Comprehensive error handling

Category Routes

Language Support

Categories support bilingual content (Arabic and English). You can:

- **Get categories in specific language:** Add `?lang=en` or `?lang=ar` to the URL
- **Get categories in user's preferred language:** The system uses the user's language preference
- **Get full bilingual data:** Use `?lang=both` to get both languages

Unique Constraints

- **Category names are unique:** No two categories can have the same name in English OR Arabic
- **Case-insensitive:** Names are stored and compared in a case-insensitive manner
- **Error handling:** Attempting to create/update a category with a duplicate name will return a specific error message indicating which language field is duplicated

Category Object

- `name` (object, required, bilingual: `{ en: "English Name", ar: "Arabic Name" }`)
- `description` (object, optional, bilingual: `{ en: "English Description", ar: "Arabic Description" }`)

- `status` (object, optional, bilingual: { en: "active/inactive", ar: "نشط/غير نشط" }, default: { en: "inactive", ar: "غير نشط" })
- `createdBy` (object, populated with creator details)
- `createdAt` (date)
- `updatedAt` (date)

1. Get Categories

- **GET** /api/categories
- **GET** /api/categories?lang=en (English only)
- **GET** /api/categories?lang=ar (Arabic only)
- **GET** /api/categories?lang=both (Both languages)
- **GET** /api/categories?status=active (Active categories only - default)
- **GET** /api/categories?status=inactive (Inactive categories only)
- **GET** /api/categories?status=active&lang=en (Active categories in English)
- **Description:** Get categories (public). By default shows only active categories.
- **Query Parameters:**
 - `lang` (string, optional, language preference: 'en', 'ar', 'both')
 - `status` (string, optional, filter by status: 'active', 'inactive' - searches both English and Arabic status fields)
- **Response:**
 - `200 OK`: List of categories

2. Create Category

- **POST** /api/categories
- **Description:** Create a new category (business, admin, or magnet_employee only).
- **Headers:** Authorization: Bearer <token>
- **Body:**
 - `name` (object, required, bilingual: { en: "English Name", ar: "Arabic Name" })
 - `description` (object, optional, bilingual: { en: "English Description", ar: "Arabic Description" })
 - `status` (object, optional, bilingual: { en: "active/inactive", ar: "نشط/غير نشط" }, default: { en: "inactive", ar: "غير نشط" })
- **Response:**
 - `201 Created`: Category info (bilingual message)
 - `400 Bad Request`: Validation errors (bilingual message)
 - `category_name_required_both_languages`: When name is missing or incomplete
 - `category_description_required_both_languages`: When description is incomplete
 - `category_status_required_both_languages`: When status is missing or incomplete
 - `category_name_en_already_exists`: When English name already exists
 - `category_name_ar_already_exists`: When Arabic name already exists
 - `invalid_category_status`: When status values are not valid

3. Update Category

- **PUT** /api/categories/:id
- **Description:** Update a category (business, admin, or magnet_employee only).

- **Headers:** Authorization: Bearer <token>
- **Body:**
 - name (object, optional, bilingual: { en: "English Name", ar: "Arabic Name" })
 - description (object, optional, bilingual: { en: "English Description", ar: "Arabic Description" })
 - status (object, optional, bilingual: { en: "active/inactive", ar: "نشط/غير نشط" })
- **Response:**
 - 200 OK: Updated category info (bilingual message)
 - 400 Bad Request: Validation errors (bilingual message)
 - category_name_required_both_languages: When name is incomplete
 - category_description_required_both_languages: When description is incomplete
 - category_status_required_both_languages: When status is incomplete
 - category_name_en_already_exists: When English name already exists
 - category_name_ar_already_exists: When Arabic name already exists
 - invalid_category_status: When status values are not valid
 - 404 Not Found: Category not found (bilingual message)
 - 403 Forbidden: Not authorized to update this category (bilingual message)

4. Delete Category

- **DELETE** /api/categories/:id
- **Description:** Delete a category (business, admin, or magnet_employee only).
- **Headers:** Authorization: Bearer <token>
- **Response:**
 - 200 OK: Category deleted (bilingual message)
 - 404 Not Found: Category not found (bilingual message)
 - 403 Forbidden: Not authorized to delete this category (bilingual message)

Category Unique Constraint Examples

Example 1: Successful Category Creation

```
POST /api/categories
{
  "name": {
    "en": "Electronics",
    "ar": "الكترونيات"
  },
  "description": {
    "en": "Electronic devices and gadgets",
    "ar": "الأجهزة الإلكترونية والأدوات"
  }
}
```

Response: 201 Created

Example 2: Duplicate English Name Error

```
POST /api/categories
{
  "name": {
    "en": "Electronics", // This already exists
    "ar": "الكترونيات جديدة"
  }
}
```

Response: 400 Bad Request

```
{
  "status": "error",
  "message": {
    "en": "Category name in English already exists",
    "ar": "اسم الفئة باللغة الإنجليزية موجود بالفعل"
  }
}
```

Example 3: Duplicate Arabic Name Error

```
POST /api/categories
{
  "name": {
    "en": "New Electronics",
    "ar": "الكترونيات" // This already exists
  }
}
```

Response: 400 Bad Request

```
{
  "status": "error",
  "message": {
    "en": "Category name in Arabic already exists",
    "ar": "اسم الفئة باللغة العربية موجود بالفعل"
  }
}
```

Example 4: Create Category with Status

```
POST /api/categories
{
  "name": {
```

```

    "en": "Electronics",
    "ar": "الكترونيات"
},
"description": {
    "en": "Electronic devices and gadgets",
    "ar": "الأجهزة الإلكترونية والأدوات"
},
"status": {
    "en": "active",
    "ar": "نشط"
}
}
}

```

Response: 201 Created

Example 5: Create Inactive Category

```

POST /api/categories
{
  "name": {
    "en": "Discontinued Products",
    "ar": "المنتجات المتوقفة"
  },
  "description": {
    "en": "Products that are no longer available",
    "ar": "المنتجات التي لم تعد متوفرة"
  },
  "status": {
    "en": "inactive",
    "ar": "غير نشط"
  }
}

```

Response: 201 Created

Example 6: Create Category with Default Status (Inactive)

```

POST /api/categories
{
  "name": {
    "en": "Test Category",
    "ar": "فئة اختبار"
  },
  "description": {
    "en": "Test description",
    "ar": "وصف اختبار"
  }
}

```

Response: 201 Created (Status will default to { "en": "inactive", "ar": "غير نشط" })

Example 7: Invalid Status Error

```
POST /api/categories
{
  "name": {
    "en": "Test Category",
    "ar": "فئة اختبار"
  },
  "status": {
    "en": "invalid_status",
    "ar": "حالة غير صحيحة"
  }
}
```

Response: 400 Bad Request

```
{
  "status": "error",
  "message": {
    "en": "Invalid category status. Must be \"active\" or \"inactive\"",
    "ar": "حالة الفئة غير صحيحة. يجب أن تكون \"نشط\" أو \"غير نشط\""
  }
}
```

Example 8: Incomplete Status Error

```
POST /api/categories
{
  "name": {
    "en": "Test Category",
    "ar": "فئة اختبار"
  },
  "status": {
    "en": "active"
    // Missing Arabic status
  }
}
```

Response: 400 Bad Request

```
{
  "status": "error",
```

```

"message": {
    "en": "Category status is required in both English and Arabic",
    "ar": "حالة الفئة مطلوبة باللغتين الإنجليزية والعربية"
}
}

```

Order Routes

Language Support

Orders support bilingual content (Arabic and English). You can:

- **Get orders in specific language:** Add `?lang=en` or `?lang=ar` to the URL
- **Get orders in user's preferred language:** The system uses the user's language preference
- **Get full bilingual data:** Use `?lang=both` to get both languages

Order Object

- `customer` (object, populated with customer details)
- `items` (array of objects, required)
 - Each item: `{ product: ObjectId, quantity: Number }`
- `shippingAddress` (ObjectId, reference to Address)
- `status` (object, bilingual: `{ en: "pending", ar: "قيد الانتظار" }`)
- `statusLog` (array of objects)
 - Each log: `{ status: { en: "pending", ar: "قيد الانتظار" }, timestamp: Date }`
- `createdAt` (date)
- `updatedAt` (date)

Note: Order status is stored as bilingual objects but returned as localized strings based on the `lang` parameter:

- `pending` → `{ en: "pending", ar: "قيد الانتظار" }`
- `confirmed` → `{ en: "confirmed", ar: "مؤكدة" }`
- `shipped` → `{ en: "shipped", ar: "تم الشحن" }`
- `delivered` → `{ en: "delivered", ar: "تم التوصيل" }`
- `cancelled` → `{ en: "cancelled", ar: "ملغى" }`

1. Create Order

- **POST** `/api/orders`
- **Description:** Create a new order (customer only).
- **Headers:** `Authorization: Bearer <token>`
- **Body:**
 - Order details (see order model)
- **Response:**
 - `201 Created`: Order info with calculated total (bilingual message)
 - **Note:** Response includes `total` and `itemTotal` fields automatically calculated from product prices and quantities

2. Get My Orders

- **GET** /api/orders/my
- **GET** /api/orders/my?lang=en (English only)
- **GET** /api/orders/my?lang=ar (Arabic only)
- **GET** /api/orders/my?lang=both (Both languages)
- **Description:** Get all orders for the authenticated customer.
- **Headers:** Authorization: Bearer <token>
- **Response:**
 - 200 OK: List of orders with calculated totals (bilingual message)
 - **Note:** Each order includes total and itemTotal fields automatically calculated

3. Get Order by ID

- **GET** /api/orders/:id
- **GET** /api/orders/:id?lang=en (English only)
- **GET** /api/orders/:id?lang=ar (Arabic only)
- **GET** /api/orders/:id?lang=both (Both languages)
- **Description:** Get order by ID (admin, magnet_employee, or owner).
- **Headers:** Authorization: Bearer <token>
- **Response:**
 - 200 OK: Order info with calculated total (bilingual message)
 - **Note:** Response includes total and itemTotal fields automatically calculated

4. Get All Orders

- **GET** /api/orders
- **GET** /api/orders?lang=en (English only)
- **GET** /api/orders?lang=ar (Arabic only)
- **GET** /api/orders?lang=both (Both languages)
- **Description:** Get all orders (admin, magnet_employee only).
- **Headers:** Authorization: Bearer <token>
- **Response:**
 - 200 OK: List of orders with calculated totals (bilingual message)
 - **Note:** Each order includes total and itemTotal fields automatically calculated

5. Update Order Status

- **PUT** /api/orders/:id/status
- **Description:** Update order status (admin, magnet_employee only).
- **Headers:** Authorization: Bearer <token>
- **Body:**
 - status (string, required, enum: 'pending', 'confirmed', 'shipped', 'delivered', 'cancelled')
- **Response:**
 - 200 OK: Updated order info with calculated total (bilingual message)
 - **Note:** Response includes total and itemTotal fields automatically calculated

6. Get Status Options

- **GET** /api/orders/status-options
- **GET** /api/orders/status-options?lang=en (English only)
- **GET** /api/orders/status-options?lang=ar (Arabic only)
- **GET** /api/orders/status-options?lang=both (Both languages)
- **Description:** Get available order status options (public).
- **Response:**

- 200 OK: Status options in requested language
- **Example Response:**

```
{
  "status": "success",
  "data": {
    "statusOptions": {
      "pending": "قيد الانتظار",
      "confirmed": "مؤكدة",
      "shipped": "تم الشحن",
      "delivered": "تم التوصيل",
      "cancelled": "ملغى"
    },
    "statusEnums": {
      "en": ["pending", "confirmed", "shipped", "delivered",
      "cancelled"],
      "ar": ["قيد الانتظار", "مؤكدة", "تم الشحن", "تم التوصيل",
      "ملغى"]
    },
    "validEnglishStatuses": ["pending", "confirmed", "shipped",
    "delivered", "cancelled"]
  }
}
```

7. Get Business Product Orders

- **GET** /api/orders/business-products
- **Description:** Get all orders containing products owned by the authenticated business user.
- **Headers:** Authorization: Bearer <token>
- **Response:**
 - 200 OK: List of orders containing the business user's products with calculated totals
 - **Note:** This endpoint returns orders that contain any product owned by the business user, with customer details for order fulfillment. Each order includes `total` and `itemTotal` fields automatically calculated.

Address Routes

Language Support

Addresses no longer support bilingual content. All address fields are stored as simple strings.

Address Object

- `user` (object, populated with user details)
- `addressLine1` (string, required)
- `addressLine2` (string, optional)
- `city` (string, required)
- `state` (string, required)
- `postalCode` (string, required)
- `country` (string, required)
- `createdAt` (date)
- `updatedAt` (date)

1. Get Addresses

- **GET** `/api/addresses`
- **Description:** Get all addresses for the authenticated customer.
- **Headers:** `Authorization: Bearer <token>`
- **Response:**
 - `200 OK`: List of addresses (bilingual message)

2. Add Address

- **POST** `/api/addresses`
- **Description:** Add a new address (customer only).
- **Headers:** `Authorization: Bearer <token>`
- **Body:**
 - `addressLine1` (string, required)
 - `addressLine2` (string, optional)
 - `city` (string, required)
 - `state` (string, required)
 - `postalCode` (string, required)
 - `country` (string, required)
- **Response:**
 - `201 Created`: Address info (bilingual message)

3. Update Address

- **PUT** `/api/addresses/:id`
- **Description:** Update an address (customer only).
- **Headers:** `Authorization: Bearer <token>`
- **Body:**
 - Any address field (see above)
- **Response:**
 - `200 OK`: Updated address info (bilingual message)

4. Delete Address

- **DELETE** `/api/addresses/:id`

- **Description:** Delete an address (customer only).
 - **Headers:** Authorization: Bearer <token>
 - **Response:**
 - 200 OK: Address deleted (bilingual message)
-

Wishlist Routes

Wishlist Object

- **user** (object, populated with wishlist owner details)
- **products** (array of objects, populated with product details including owner and approvedBy)
- **createdAt** (date)

1. Get Wishlist

- **GET** /api/wishlist
- **Description:** Get the authenticated customer's wishlist.
- **Headers:** Authorization: Bearer <token>
- **Response:**
 - 200 OK: List of wishlist items (bilingual message)

2. Add to Wishlist

- **POST** /api/wishlist
- **Description:** Add a product to the authenticated customer's wishlist.
- **Headers:** Authorization: Bearer <token>
- **Body:**
 - **productId** (string, required)
- **Response:**
 - 200 OK: Wishlist item info (bilingual message)

3. Remove from Wishlist

- **DELETE** /api/wishlist/:productId
 - **Description:** Remove a product from the authenticated customer's wishlist.
 - **Headers:** Authorization: Bearer <token>
 - **Response:**
 - 200 OK: Wishlist item removed (bilingual message)
-

Review Routes

Language Support

Reviews no longer support bilingual content. All review comments are stored as simple strings.

Review Object

- **product** (object, populated with product details including owner and approvedBy)

- `user` (object, populated with reviewer details)
- `rating` (number, required, min: 1, max: 5)
- `comment` (string, optional)
- `status` (string, enum: 'accept', 'reject', default: 'accept')
- `rejectedBy` (object, populated with admin details who rejected, only if status is 'reject')
- `rejectedAt` (date, only if status is 'reject')
- `rejectionReason` (string, only if status is 'reject')
- `createdAt` (date)

1. Add Review

- **POST** `/api/reviews/products/:id/reviews`
- **Description:** Add a review to a product (customer only).
- **Headers:** `Authorization: Bearer <token>`
- **Body:**
 - `rating` (number, required, min: 1, max: 5)
 - `comment` (string, optional)
- **Response:**
 - `201 Created`: Review info (bilingual message)

2. Get Product Reviews

- **GET** `/api/reviews/products/:id/reviews`
- **Description:** Get all reviews for a product (public).
- **Response:**
 - `200 OK`: List of reviews (bilingual message)

3. Get Business Products Reviews

- **GET** `/api/reviews/business-products-reviews`
- **Description:** Get all reviews for products owned by the authenticated business user.
- **Headers:** `Authorization: Bearer <token>`
- **Response:**
 - `200 OK`: List of reviews for the business user's products
 - **Note:** This endpoint returns reviews for all products owned by the business user, with user and product details.

4. Delete Review

- **DELETE** `/api/reviews/:id`
- **Description:** Delete a review (admin or magnet_employee only).
- **Headers:** `Authorization: Bearer <token>`
- **Response:**
 - `200 OK`: Review deleted (bilingual message)

User Routes

1. Get User Profile

- **GET** /api/user/profile
- **Description:** Get the authenticated user's profile.
- **Headers:** Authorization: Bearer <token>
- **Response:**
 - 200 OK: User profile info (bilingual message)

2. Update User Profile

- **PUT** /api/user/profile
- **Description:** Update the authenticated user's profile.
- **Headers:** Authorization: Bearer <token>
- **Body:**
 - firstname (string, optional)
 - lastname (string, optional)
 - email (string, optional)
 - phone (string, optional)
 - country (string, optional)
 - language (string, optional)
- **Response:**
 - 200 OK: Updated user profile (bilingual message)

3. Get Business Requests

- **GET** /api/user/business-requests
- **Description:** Get all business registration requests (admin/employee only).
- **Headers:** Authorization: Bearer <token>
- **Response:**
 - 200 OK: List of business requests (bilingual message)

4. Business Approval

- **POST** /api/user/business-approval
- **Description:** Approve or reject business registration (admin/employee only).
- **Headers:** Authorization: Bearer <token>
- **Body:**
 - businessId (string, required)
 - status (string, required, enum: 'approved', 'rejected')
 - reason (string, optional, required if status is 'rejected')
- **Response:**
 - 200 OK: Business approval status updated (bilingual message)

5. Get Business Details

- **GET** /api/user/business/:businessId
- **Description:** Get business details by ID (admin/employee only).
- **Headers:** Authorization: Bearer <token>
- **Response:**

- 200 OK: Business details (bilingual message)

6. Get Business Profile

- **GET /api/user/business-profile**
 - **Description:** Get business profile (business users only).
 - **Headers:** Authorization: Bearer <token>
 - **Response:**
 - 200 OK: Business profile info (bilingual message)
-

Admin Routes

Overview

Admin routes provide comprehensive user management functionality for system administrators. All admin endpoints require authentication and admin role authorization.

Key Features:

- **User Management:** Create, read, update, delete users (admin only)
- **User Status Control:** Allow/disallow users from accessing the system (admin only)
- **Verification Management:** Manually verify/unverify user email and phone numbers (admin or magnet_employee)
- **Business Approval:** Manage business user approvals and rejections
- **Statistics:** Get comprehensive user statistics for dashboard
- **Role-based Access:** Support for all user roles (customer, business, magnet_employee, admin)
- **Enhanced Data:** All APIs return complete user information with populated related fields

1. Create User

- **POST /api/admin/users**
- **Description:** Create any type of user (customer, business, magnet_employee, admin).
- **Headers:** Authorization: Bearer <token>
- **Body:**
 - **firstname** (string, required)
 - **lastname** (string, required)
 - **email** (string, required)
 - **phone** (string, optional)
 - **password** (string, required)
 - **role** (string, required, enum: 'customer', 'business', 'magnet_employee', 'admin')
 - **country** (string, required)
 - **language** (string, optional, default: 'en')
 - **imageUrl** (string, optional) - Profile image URL
 - **businessInfo** (object, required if role is 'business')
 - **crNumber** (string, required for business)
 - **vatNumber** (string, required for business)
 - **companyName** (string, required for business)
 - **companyType** (string, required for business)

- `city` (string, required for business)
- `district` (string, required for business)
- `streetName` (string, required for business)

- **Response:**

- `201 Created`: User created successfully (bilingual message)
- `400 Bad Request`: Validation errors (bilingual message)
- `403 Forbidden`: Insufficient permissions (bilingual message)

2. Get All Users

- **GET** `/api/admin/users`
- **Description:** Get all users with pagination, filtering, and search capabilities. Automatically fixes missing `approvedBy` field for approved business users.
- **Headers:** `Authorization: Bearer <token>`
- **Query Parameters:**
 - `page` (number, optional, default: 1)
 - `limit` (number, optional, default: 10)
 - `role` (string, optional, filter by role)
 - `search` (string, optional, search by name, email, or phone)
 - `status` (string, optional, filter by disallow status: 'allowed', 'disallowed')
- **Response:**
 - `200 OK`: List of users with pagination info (bilingual message)

3. Get User Statistics

- **GET** `/api/admin/users/stats`
- **Description:** Get comprehensive user statistics for the admin dashboard.
- **Headers:** `Authorization: Bearer <token>`
- **Response:**
 - `200 OK`: User statistics including counts by role, verification status, etc. (bilingual message)

4. Get User by ID

- **GET** `/api/admin/users/:id`
- **Description:** Get detailed information about a specific user. Automatically fixes missing `approvedBy` field for approved business users. For business users, returns products and their reviews. For customer users, returns orders, wishlist, addresses, and reviews. All products include their first image URL.
- **Headers:** `Authorization: Bearer <token>`
- **Response:**
 - `200 OK`: User details with role-specific data (bilingual message)
 - `404 Not Found`: User not found (bilingual message)
- **Business User Response:**
 - `products`: Array of products with `imageUrl` (first image from product's images array)
 - `reviews`: Array of reviews with product details including `imageUrl`
- **Customer User Response:**
 - `orders`: Array of orders with product details including `imageUrl`
 - `wishlist`: Wishlist with products including `imageUrl`
 - `addresses`: Array of addresses

- `reviews`: Array of reviews with product details including `imageUrl`
- **Admin/Magnet Employee User Response:**
 - `moderatedProducts`: Array of products approved/declined by this admin/employee with owner details and `imageUrl`
 - `rejectedReviews`: Array of reviews rejected by this admin/employee with user and product details including `imageUrl`
 - `moderatedOrders`: Array of orders where this admin/employee made status changes (confirmed, shipped, cancelled, etc.)
 - `approvedBusinessUsers`: Array of business users approved by this admin/employee with complete business information
- **Product Object in Responses:**
 - `id`: Product ID
 - `name`: Product name (bilingual object)
 - `code`: Product code
 - `status`: Product status
 - `category`: Product category (bilingual object)
 - `imageUrl`: First image URL from product's images array (null if no images)

5. Update User

- **PUT** `/api/admin/users/:id`
- **Description:** Update user information (cannot update own account).
- **Headers:** `Authorization: Bearer <token>`
- **Body:**
 - `firstname` (string, optional)
 - `lastname` (string, optional)
 - `email` (string, optional)
 - `phone` (string, optional)
 - `password` (string, optional)
 - `role` (string, optional, enum: 'customer', 'business', 'magnet_employee', 'admin')
 - `country` (string, optional)
 - `language` (string, optional)
 - `imageUrl` (string, optional) - Profile image URL
 - `isEmailVerified` (boolean, optional)
 - `isPhoneVerified` (boolean, optional)
 - `isDisallowed` (boolean, optional)
 - `businessInfo` (object, optional for business users)
 - `crNumber` (string, optional)
 - `vatNumber` (string, optional)
 - `companyName` (string, optional)
 - `companyType` (string, optional)
 - `city` (string, optional)
 - `district` (string, optional)
 - `streetName` (string, optional)
- **Response:**
 - `200 OK`: User updated successfully (bilingual message)
 - `400 Bad Request`: Validation errors (bilingual message)

- 404 Not Found: User not found (bilingual message)

6. Delete User

- **DELETE** /api/admin/users/:id
- **Description:** Permanently delete a user (cannot delete own account).
- **Headers:** Authorization: Bearer <token>
- **Response:**
 - 200 OK: User deleted successfully (bilingual message)
 - 400 Bad Request: Cannot delete own account (bilingual message)
 - 404 Not Found: User not found (bilingual message)

7. Disallow User

- **PUT** /api/admin/users/:id/disallow
- **Description:** Disallow a user from logging in and performing actions.
- **Headers:** Authorization: Bearer <token>
- **Body:**
 - reason (string, required) - Reason for disallowing the user
- **Response:**
 - 200 OK: User disallowed successfully (bilingual message)
 - 400 Bad Request: Cannot disallow admin or own account (bilingual message)
 - 404 Not Found: User not found (bilingual message)

8. Allow User

- **PUT** /api/admin/users/:id/allow
- **Description:** Allow a previously disallowed user to login and perform actions.
- **Headers:** Authorization: Bearer <token>
- **Response:**
 - 200 OK: User allowed successfully (bilingual message)
 - 400 Bad Request: User is not disallowed (bilingual message)
 - 404 Not Found: User not found (bilingual message)

9. Verify User Email

- **PUT** /api/admin/users/:id/verify-email
- **Description:** Manually verify a user's email address.
- **Headers:** Authorization: Bearer <token>
- **Response:**
 - 200 OK: Email verified successfully (bilingual message)
 - 400 Bad Request: Email is already verified (bilingual message)
 - 404 Not Found: User not found (bilingual message)
- **Example Response:**

```
{  
  "status": "success",  
  "data": {
```

```

"user": {
    "id": "user_id",
    "firstname": "John",
    "lastname": "Doe",
    "email": "john@example.com",
    "isEmailVerified": true,
    "isPhoneVerified": false
},
},
"message": {
    "en": "Email verified successfully",
    "ar": "تم التحقق من البريد الإلكتروني بنجاح"
}
}

```

10. Unverify User Email

- **PUT /api/admin/users/:id/unverify-email**
- **Description:** Manually unverify a user's email address.
- **Headers:** Authorization: Bearer <token>
- **Response:**
 - **200 OK:** Email unverified successfully (bilingual message)
 - **400 Bad Request:** Email is not verified (bilingual message)
 - **404 Not Found:** User not found (bilingual message)
- **Example Response:**

```

{
    "status": "success",
    "data": {
        "user": {
            "id": "user_id",
            "firstname": "John",
            "lastname": "Doe",
            "email": "john@example.com",
            "isEmailVerified": false,
            "isPhoneVerified": false
        }
    },
    "message": {
        "en": "Email unverified successfully",
        "ar": "تم إلغاء التحقق من البريد الإلكتروني بنجاح"
    }
}

```

11. Verify User Phone

- **PUT /api/admin/users/:id/verify-phone**
- **Description:** Manually verify a user's phone number.
- **Headers:** Authorization: Bearer <token>

- **Response:**

- 200 OK: Phone verified successfully (bilingual message)
- 400 Bad Request: Phone is already verified or user has no phone (bilingual message)
- 404 Not Found: User not found (bilingual message)

- **Example Response:**

```
{
  "status": "success",
  "data": {
    "user": {
      "id": "user_id",
      "firstname": "John",
      "lastname": "Doe",
      "phone": "+1234567890",
      "isEmailVerified": true,
      "isPhoneVerified": true
    }
  },
  "message": {
    "en": "Phone verified successfully",
    "ar": "تم التحقق من الهاتف بنجاح"
  }
}
```

12. Unverify User Phone

- **PUT** /api/admin/users/:id/unverify-phone
- **Description:** Manually unverify a user's phone number.
- **Headers:** Authorization: Bearer <token>
- **Response:**
 - 200 OK: Phone unverified successfully (bilingual message)
 - 400 Bad Request: Phone is not verified (bilingual message)
 - 404 Not Found: User not found (bilingual message)

- **Example Response:**

```
{
  "status": "success",
  "data": {
    "user": {
      "id": "user_id",
      "firstname": "John",
      "lastname": "Doe",
      "phone": "+1234567890",
      "isEmailVerified": true,
      "isPhoneVerified": false
    }
  },
  "message": {
    "en": "Phone unverified successfully",
    "ar": "تم الغاء التحقق من الهاتف بنجاح"
  }
}
```

```
        "en": "Phone unverified successfully",
        "ar": "تم إلغاء التحقق من الهاتف بنجاح"
    }
}
```

Verification Management Features

The admin verification management system provides comprehensive control over user verification status:

Email Verification:

- **Verify Email:** Manually mark a user's email as verified
- **Unverify Email:** Remove email verification status
- **Validation:** Prevents duplicate verification or unverification of already unverified emails

Phone Verification:

- **Verify Phone:** Manually mark a user's phone as verified
- **Unverify Phone:** Remove phone verification status
- **Validation:**
 - Prevents verification if user has no phone number
 - Prevents duplicate verification or unverification of already unverified phones

Security Features:

- **Admin or Magnet Employee:** All verification endpoints require admin or magnet_employee authentication
- **Audit Trail:** All verification changes are tracked with timestamps
- **Bilingual Support:** All responses include English and Arabic messages
- **Error Handling:** Comprehensive validation with appropriate error messages

Use Cases:

- **Manual Verification:** Admins can verify users who completed verification through other means
- **Account Recovery:** Help users regain access by re-verifying their contact information
- **Security Management:** Unverify suspicious accounts for security review
- **Compliance:** Ensure verification status meets regulatory requirements

Admin User Object

- **id** (ObjectId)
- **firstname** (string)
- **lastname** (string)
- **email** (string)
- **phone** (string, optional)
- **role** (string, enum: 'admin', 'magnet_employee', 'business', 'customer')
- **country** (string)
- **language** (string, enum: 'en', 'ar')
- **imageUrl** (string, optional)
- **isEmailVerified** (boolean)

- `isPhoneVerified` (boolean)
 - `isDisallowed` (boolean)
 - `disallowReason` (string, if disallowed)
 - `disallowedBy` (object, populated with admin details who disallowed)
 - `disallowedAt` (Date, if disallowed)
 - `allowedBy` (object, populated with admin details who allowed)
 - `allowedAt` (Date, if allowed)
 - `businessInfo` (object, for business users)
 - `crNumber` (string)
 - `vatNumber` (string)
 - `companyName` (string)
 - `companyType` (string)
 - `city` (string)
 - `district` (string)
 - `streetName` (string)
 - `isApproved` (boolean)
 - `approvalStatus` (string, enum: 'pending', 'approved', 'rejected')
 - `approvedBy` (object, populated with admin details who approved)
 - `approvedAt` (Date)
 - `rejectionReason` (string, if rejected)
 - `createdAt` (Date)
 - `updatedAt` (Date)
-

Admin Wishlist Management Routes

1. Get All Wishlists

- **GET** `/api/admin/wishlists`
- **Description:** Get all wishlists with pagination and filtering capabilities.
- **Headers:** `Authorization: Bearer <token>`
- **Query Parameters:**
 - `page` (number, optional, default: 1)
 - `limit` (number, optional, default: 10)
 - `userId` (string, optional, filter by user ID)
 - `productId` (string, optional, filter by product ID)
- **Response:**
 - `200 OK`: List of wishlists with pagination info (bilingual message)
 - **Example Response:**

```
{  
  "status": "success",  
  "data": {  
    "wishlists": [  
      {  
        "id": "wishlist_id",  
        "user": {  
          "name": "John Doe",  
          "email": "john.doe@example.com",  
          "created_at": "2023-08-28T12:00:00Z",  
          "updated_at": "2023-08-28T12:00:00Z"  
        }  
      }  
    ]  
  }  
}
```

```

        "id": "user_id",
        "firstname": "John",
        "lastname": "Doe",
        "email": "john@example.com",
        "role": "customer"
    },
    "products": [
        {
            "id": "product_id",
            "name": "Product Name",
            "code": "PROD001",
            "status": "approved"
        }
    ],
    "createdAt": "2024-01-01T00:00:00.000Z"
}
],
},
"pagination": {
    "currentPage": 1,
    "totalPages": 1,
    "totalItems": 1,
    "itemsPerPage": 10
}
}

```

2. Get Wishlist by ID

- **GET /api/admin/wishlists/:id**
- **Description:** Get detailed information about a specific wishlist.
- **Headers:** Authorization: Bearer <token>
- **Response:**
 - 200 OK: Wishlist details (bilingual message)
 - 404 Not Found: Wishlist not found (bilingual message)

3. Create Wishlist

- **POST /api/admin/wishlists**
- **Description:** Add a product to a user's wishlist (creates wishlist if it doesn't exist).
- **Headers:** Authorization: Bearer <token>
- **Body:**
 - **userId** (string, required) - User ID
 - **productId** (string, required) - Product ID (must be approved)
- **Response:**
 - 201 Created: Product added to wishlist successfully (bilingual message)
 - 400 Bad Request: Validation errors (bilingual message)

4. Update Wishlist

- **PUT /api/admin/wishlists/:id**

- **Description:** Update wishlist information or add/remove products.
- **Headers:** Authorization: Bearer <token>
- **Body:**
 - userId (string, optional) - New user ID
 - productId (string, optional) - Product ID to add or remove
 - action (string, optional, enum: 'add', 'remove') - Action to perform on product
- **Response:**
 - 200 OK: Wishlist updated successfully (bilingual message)
 - 404 Not Found: Wishlist not found (bilingual message)

5. Delete Wishlist

- **DELETE** /api/admin/wishlists/:id
 - **Description:** Permanently delete a wishlist entry.
 - **Headers:** Authorization: Bearer <token>
 - **Response:**
 - 200 OK: Wishlist deleted successfully (bilingual message)
 - 404 Not Found: Wishlist not found (bilingual message)
-

Admin Review Management Routes

1. Get All Reviews

- **GET** /api/admin/reviews
- **Description:** Get all reviews with pagination and filtering capabilities.
- **Headers:** Authorization: Bearer <token>
- **Query Parameters:**
 - page (number, optional, default: 1)
 - limit (number, optional, default: 10)
 - userId (string, optional, filter by user ID)
 - productId (string, optional, filter by product ID)
 - rating (number, optional, filter by rating)
- **Response:**
 - 200 OK: List of reviews with pagination info (bilingual message)

2. Get Review by ID

- **GET** /api/admin/reviews/:id
- **Description:** Get detailed information about a specific review.
- **Headers:** Authorization: Bearer <token>
- **Response:**
 - 200 OK: Review details (bilingual message)
 - 404 Not Found: Review not found (bilingual message)

3. Create Review

- **POST** /api/admin/reviews
- **Description:** Create a new review for any user and product.

- **Headers:** Authorization: Bearer <token>
- **Body:**
 - userId (string, required) - User ID
 - productId (string, required) - Product ID (must be approved)
 - rating (number, required, min: 1, max: 5) - Rating
 - comment (string, optional) - Review comment
- **Response:**
 - 201 Created: Review created successfully (bilingual message)
 - 400 Bad Request: Validation errors (bilingual message)

4. Update Review

- **PUT** /api/admin/reviews/:id
- **Description:** Update review information.
- **Headers:** Authorization: Bearer <token>
- **Body:**
 - rating (number, optional, min: 1, max: 5) - New rating
 - comment (string, optional) - New comment
- **Response:**
 - 200 OK: Review updated successfully (bilingual message)
 - 404 Not Found: Review not found (bilingual message)

5. Delete Review

- **DELETE** /api/admin/reviews/:id
- **Description:** Permanently delete a review.
- **Headers:** Authorization: Bearer <token>
- **Response:**
 - 200 OK: Review deleted successfully (bilingual message)
 - 404 Not Found: Review not found (bilingual message)

6. Reject Review

- **PUT** /api/admin/reviews/:id/reject
- **Description:** Reject a review and send email notification to the reviewer.
- **Headers:** Authorization: Bearer <token>
- **Body:**
 - reason (string, required) - Reason for rejecting the review
- **Response:**
 - 200 OK: Review rejected successfully (bilingual message)
 - 400 Bad Request: Review already rejected or missing rejection reason (bilingual message)
 - 404 Not Found: Review not found (bilingual message)
- **Features:**
 - **Email Notification:** Automatically sends email to the reviewer informing them of the rejection
 - **Status Tracking:** Updates review status to 'reject' and tracks rejection details
 - **Audit Trail:** Records who rejected the review, when, and why
 - **Validation:** Prevents rejecting already rejected reviews
- **Example Response:**

```
{
  "status": "success",
  "data": {
    "review": {
      "id": "review_id",
      "user": {
        "id": "user_id",
        "firstname": "John",
        "lastname": "Doe",
        "email": "john@example.com"
      },
      "product": {
        "id": "product_id",
        "name": "Product Name"
      },
      "rating": 4,
      "comment": "Great product!",
      "status": "reject",
      "rejectedBy": {
        "id": "admin_id",
        "firstname": "Admin",
        "lastname": "User",
        "email": "admin@example.com",
        "role": "admin"
      },
      "rejectedAt": "2024-01-01T12:00:00.000Z",
      "rejectionReason": "Inappropriate content",
      "createdAt": "2024-01-01T10:00:00.000Z"
    }
  },
  "message": {
    "en": "Review rejected successfully",
    "ar": "تم رفض التقييم بنجاح"
  }
}
```

Admin Address Management Routes

1. Get All Addresses

- **GET** /api/admin/addresses
- **Description:** Get all addresses with pagination and filtering capabilities.
- **Headers:** Authorization: Bearer <token>
- **Query Parameters:**
 - page (number, optional, default: 1)
 - limit (number, optional, default: 10)
 - userName (string, optional, search by user name, email, or phone - supports partial matches and multiple search patterns)
 - city (string, optional, search by city)

- **country** (string, optional, search by country)
- **Search Functionality:**
 - The **userName** parameter performs a comprehensive search across:
 - User first name (exact match, contains, starts with, ends with)
 - User last name (exact match, contains, starts with, ends with)
 - User email address
 - User phone number
 - Full name (concatenated firstname + lastname)
 - Search is case-insensitive and supports partial matching
 - Examples: `?userName=john` will find "John", "Johnny", "Johnson", etc.
- **Response:**
 - **200 OK**: List of addresses with pagination info (bilingual message)

2. Get Address by ID

- **GET** `/api/admin/addresses/:id`
- **Description:** Get detailed information about a specific address.
- **Headers:** `Authorization: Bearer <token>`
- **Response:**
 - **200 OK**: Address details (bilingual message)
 - **404 Not Found**: Address not found (bilingual message)

3. Create Address

- **POST** `/api/admin/addresses`
- **Description:** Create a new address for any user.
- **Headers:** `Authorization: Bearer <token>`
- **Body:**
 - **userId** (string, required) - User ID
 - **addressLine1** (string, required) - Primary address line
 - **addressLine2** (string, optional) - Secondary address line
 - **city** (string, required) - City
 - **state** (string, required) - State/Province
 - **postalCode** (string, required) - Postal code
 - **country** (string, required) - Country
- **Response:**
 - **201 Created**: Address created successfully (bilingual message)
 - **400 Bad Request**: Validation errors (bilingual message)

4. Update Address

- **PUT** `/api/admin/addresses/:id`
- **Description:** Update address information.
- **Headers:** `Authorization: Bearer <token>`
- **Body:**
 - **addressLine1** (string, optional) - Primary address line
 - **addressLine2** (string, optional) - Secondary address line
 - **city** (string, optional) - City

- `state` (string, optional) - State/Province
- `postalCode` (string, optional) - Postal code
- `country` (string, optional) - Country
- **Response:**
 - `200 OK`: Address updated successfully (bilingual message)
 - `404 Not Found`: Address not found (bilingual message)

5. Delete Address

- **DELETE** `/api/admin/addresses/:id`
- **Description:** Permanently delete an address.
- **Headers:** `Authorization: Bearer <token>`
- **Response:**
 - `200 OK`: Address deleted successfully (bilingual message)
 - `404 Not Found`: Address not found (bilingual message)

Admin Addresses Search Examples

Example 1: Search by User Name

```
GET /api/admin/addresses?userName=hussien
```

Description: Returns all addresses for users with "hussien" in their first name, last name, or full name

Example 2: Search by Email

```
GET /api/admin/addresses?userName=adham@gmail.com
```

Description: Returns all addresses for users with this email address

Example 3: Search by Phone

```
GET /api/admin/addresses?userName=+966501234567
```

Description: Returns all addresses for users with this phone number

Example 4: Search with City Filter

```
GET /api/admin/addresses?userName=hussien&city=aswan
```

Description: Returns addresses for users named "hussien" in the city of "aswan"

Example 5: Search with Pagination

```
GET /api/admin/addresses?userName=john&page=1&limit=5
```

Description: Returns first 5 addresses for users with "john" in their name

Example 6: Combined Filters

```
GET /api/admin/addresses?userName=hussien&city=aswan&country=egypt&page=1&limit=10
```

Description: Returns first 10 addresses for users named "hussien" in "aswan" city, "egypt" country

Admin Order Management Routes

1. Get All Orders

- **GET /api/admin/orders**
- **Description:** Get all orders with pagination and filtering capabilities.
- **Headers:** Authorization: Bearer <token>
- **Query Parameters:**
 - page (number, optional, default: 1)
 - limit (number, optional, default: 10)
 - customerName (string, optional, search by customer name, email, or phone - supports partial matches and multiple search patterns)
 - status (string, optional, filter by order status)
 - date (string, optional, filter orders created on specific date - format: YYYY-MM-DD)
 - lang (string, optional, language preference: 'en', 'ar', 'both')
- **Search Functionality:**
 - The customerName parameter performs a comprehensive search across:
 - Customer first name (exact match, contains, starts with, ends with)
 - Customer last name (exact match, contains, starts with, ends with)
 - Customer email address
 - Customer phone number
 - Full name (concatenated firstname + lastname)
 - Search is case-insensitive and supports partial matching
 - Examples: ?customerName=john will find "John", "Johnny", "Johnson", etc.
- **Response:**
 - 200 OK: List of orders with pagination info and calculated totals (bilingual message)
 - **Note:** Each order includes total and itemTotal fields automatically calculated from product prices and quantities

Admin Orders Date Filtering Examples

Example 1: Get Orders for a Specific Date

```
GET /api/admin/orders?date=2025-08-27
```

Description: Returns all orders created on August 27, 2025 (from 00:00:00 to 23:59:59)

Example 2: Get Orders for a Date with Status Filter

```
GET /api/admin/orders?date=2025-08-27&status=pending
```

Description: Returns all pending orders created on August 27, 2025

Example 3: Get Orders for a Date with Customer Search

```
GET /api/admin/orders?date=2025-08-27&customerName=hussien
```

Description: Returns all orders for customer "hussien" created on August 27, 2025

Example 4: Get Orders for a Date with Pagination

```
GET /api/admin/orders?date=2025-08-27&page=1&limit=5
```

Description: Returns first 5 orders created on August 27, 2025

Example 5: Combined Filters

```
GET /api/admin/orders?date=2025-08-27&status=pending&customerName=hussien&page=1&limit=10
```

Description: Returns first 10 pending orders for customer "hussien" created on August 27, 2025

Date Format Notes

- **Supported format:** YYYY-MM-DD (ISO date format)
- **Examples:** 2025-08-27, 2025-8-27 (both work)
- **Time range:** The filter covers the entire day from 00:00:00 to 23:59:59
- **Timezone:** Uses UTC timezone for date comparisons

2. Get Order by ID

- **GET /api/admin/orders/:id**
- **Description:** Get detailed information about a specific order.

- **Headers:** Authorization: Bearer <token>
- **Query Parameters:**
 - lang (string, optional, language preference: 'en', 'ar', 'both')
- **Response:**
 - 200 OK: Order details with calculated total (bilingual message)
 - 404 Not Found: Order not found (bilingual message)
 - **Note:** Response includes total and itemTotal fields automatically calculated

3. Create Order

- **POST** /api/admin/orders
- **Description:** Create a new order for any customer.
- **Headers:** Authorization: Bearer <token>
- **Body:**
 - customerId (string, required) - Customer ID
 - items (array, required) - Array of order items
 - Each item: { product: string, quantity: number }
 - shippingAddressId (string, required) - Shipping address ID
- **Query Parameters:**
 - lang (string, optional, language preference: 'en', 'ar', 'both')
- **Response:**
 - 201 Created: Order created successfully with calculated total (bilingual message)
 - 400 Bad Request: Validation errors (bilingual message)
 - **Note:** Response includes total and itemTotal fields automatically calculated from product prices and quantities

4. Update Order

- **PUT** /api/admin/orders/:id
- **Description:** Update order information including status.
- **Headers:** Authorization: Bearer <token>
- **Body:**
 - items (array, optional) - New order items
 - shippingAddressId (string, optional) - New shipping address ID
 - status (string, optional) - New order status
- **Query Parameters:**
 - lang (string, optional, language preference: 'en', 'ar', 'both')
- **Response:**
 - 200 OK: Order updated successfully with calculated total (bilingual message)
 - 404 Not Found: Order not found (bilingual message)
 - **Note:** Response includes total and itemTotal fields automatically calculated when items are updated

5. Delete Order

- **DELETE** /api/admin/orders/:id
- **Description:** Permanently delete an order.
- **Headers:** Authorization: Bearer <token>

- **Response:**
 - **200 OK:** Order deleted successfully (bilingual message)
 - **404 Not Found:** Order not found (bilingual message)
-

Changelog

Latest Updates (2025-01-27)

Category Management Enhancements

- **Enhanced Status Field:** Category status is now bilingual (`{ en: "active/inactive", ar: "نشط/غير نشط" }`) with default: `{ en: "inactive", ar: "غير نشط" }`
- **Added Unique Constraint:** Category names are now unique across both English and Arabic languages
- **Enhanced Error Handling:** Specific error messages for duplicate category names and incomplete status fields
- **Improved Validation:** Better validation for bilingual category fields including status values
- **Updated API Responses:** Added detailed error responses for create and update operations
- **Status Filtering:** GET categories now filters by bilingual status (default: active only)

Admin Addresses API Improvements

- **Enhanced User Search:** Replaced `userId` parameter with `userName` for comprehensive user search
- **Multi-field Search:** Search now works across user first name, last name, email, and phone
- **Pattern Matching:** Supports exact match, contains, starts with, and ends with patterns
- **Case-insensitive:** All searches are case-insensitive for better user experience

Admin Orders API Improvements

- **Simplified Date Filtering:** Replaced `startDate` and `endDate` parameters with single `date` parameter
- **Better Date Range:** Date filter now covers entire day (00:00:00 to 23:59:59)
- **Enhanced Examples:** Added comprehensive examples for date filtering with various combinations
- **Improved Documentation:** Updated query parameter descriptions and added usage examples

Technical Improvements

- **Database Indexes:** Added unique indexes for category names in both languages
- **Error Code Handling:** Proper handling of MongoDB duplicate key errors (code 11000)
- **Bilingual Messages:** Added new error messages for duplicate category names and invalid status
- **API Consistency:** Maintained consistent error response format across all endpoints
- **Search Optimization:** Improved search functionality with multiple pattern matching

Breaking Changes

- **Admin Orders API:** The `startDate` and `endDate` parameters have been replaced with a single `date` parameter

- **⚠ Admin Addresses API:** The `userId` parameter has been replaced with `userName` for enhanced search functionality
- **⚠ Category Names:** Category names are now enforced as unique across both languages
- **⚠ Category Status:** Category status is now bilingual and defaults to `{ en: "inactive", ar: "غير نشط" }`